NATIONAL UNIVERSITY OF SINGAPORE

CS1010 – PROGRAMMING METHODOLOGY

(AY2017/18 Semester 4)

Time Allowed: 2 Hours

INSTRUCTIONS TO STUDENTS

1. This assessment paper consists of **TWELVE (12)** questions and comprises EIGHT **(8)** printed pages.
2. This is an **OPEN BOOK** assessment.
3. Calculators and electronic dictionaries are not allowed.
4. Answer all questions, and write your answers in the **ANSWER SHEETS** provided.
5. Fill in your Student Number with a pen, clearly on the first page of your ANSWER SHEETS.
6. Do NOT write your name on your ANSWER SHEETS.
7. You may use **2B pencil** to write your programs.
8. Note that there will be penalty for programs that are unclear or unnecessarily long.
9. You must submit only the ANSWER SHEETS and no other document.

Questions 7 to 12:

Write your answers in the space provided on the **Answer Sheets**. You may write in pencil. You are to write legibly or marks might be deducted.

Q7. [6 marks]

What is the output of the following program?

```
#include <stdio.h>
void f(int *, int);

int main(void){
    int a = 2, b = 3;
    f(&a, b);
    printf("%d %d\n", a, b);
    return 0;
}

void f(int *i, int j){
    int a;
    for (a = 0; a<10; a += 2){
        (*i)++;
        --j;
    }
}
```

Q8. [6 marks]

Complete the following code that reverse a string. Fill in the missing parts (a)-(d) without changing the rest of the code

```
#include <string.h>

void reverse(char* str)
{
    int length = (a) ;
    int i;
    for (i = 0; i < (b) ; i++) {
        char ch = *(str + i);
        (c) ;
        (d) = ch;
    }
}
```

Q9.[12 marks]

A three digits number is a "daffodil" number if the sum of the cube of each digit is equal to itself. For example, 153 is a daffodil number because $1^3 + 5^3 + 3^3 = 153$. Write **two C programs, one using loops and one using recursion,** to print all daffodil numbers in the range of [100, 999].

Q10. [12 marks]

An international company hired several new employees from different parts of the world to work on a project. Since the new employees are from all over the world, they have different working hours. Each employee has a name, a start working time, and an end working time.

1. **[3 marks]** Define a structure type **employee_t** for the employees, where the start working time and end working time are integer. The name of each employee consists of less than 20 characters.
2. **[3 marks]** Write a function **int read_employee(employee_t emp[])** to read the number of employees and the employees into an array. The function will return the number of employees after the input. You can assume that the number of employees is at least 1 and at most 10. An example of the input data of 5 employees is displayed as follows.

```
Enter the number of employees: 5
Alice 2 8
Bob 6 13
Helen 9 19
Jim 12 20
Zoey 15 22
```

3. **[6 marks]** In order to improve the working efficiency by ensuring that the project can be continued to work on 24 hours around the clock, the employer would like to hire a new employee such that his/her working hours has the least overlap with the working hours of all the other employees.

   Given a new employee, write a function **void overlap(employee_t new_emp, employee_t emp[], int size)** to list all existing employees whose working hours has overlap with the new employee, and return the total hours of overlap with these.

   For example, if the new employee is "Tracy 13 21", the list of overlapping employees is as follow

```
Helen 9 19
Jim 12 20
Zoey 15 22
```

   And the total overlapping time is 18.

Q11. [12 marks]

A and B are two super stars who are having an underground relationship. They cannot afford to let public knows about what is going on, so they want to communicate in an encrypted manner. The rule for encryption is shown as followed:

1. Each alphabet a, b, c, d, ... , z, A, B, C, ..., Z correspond to a number 1, 2, 3, ..., 52 respectively.
2. An alphabet is encrypted into another alphabet by
   (i) Multiplying its corresponding number by 3 to obtain **result**
   (ii) Generate the alphabet that correspond to **result** % 52

   For example, a -> (1*3 % 52) = 3 -> c; Corresponding, the sentence "IloveU" get encrypted to "aJSnoK".

a) **[4 marks]** write a function **encode** that helps them to encrypt their message.

```
char* encode(char prim[])
```

b) **[8 marks]** write a function **decode** that helps them to decrypt their message.

```
char* decode(char code[])
```

You can assume the message contains only alphabets without space.

Q12. [14 marks] Binary Insertion Sort

Beside two basic sorting algorithms (selection sort and bubble sort) which are introduced in the lecture notes, there is another simple sorting algorithm called **Insertion Sort**. Generally, the insertion sort algorithm works the way we sort playing cards in our hands. Its key idea can be illustrated using following example.

Given an array A of integers, let us loop for i = 1 (second element of the array) to 5 (size of input array)

12	10	17	4	7

- i = 1: Since 10 is smaller than 12, move 12 and insert 10 before 12;

10	**12**	17	4	7

- i = 2: 17 will remain at its position as all elements in A[0..i-1] are smaller than 17;

10	**12**	**17**	4	7

- i = 3: 4 will move the beginning and all other elements 10, 12, and 17 will move one position ahead of their current position;

4	**10**	**12**	**17**	7

- i = 4: 7 will move to position after 4, and elements 10, 12, and 17 will move one position ahead of their current position;

4	**7**	**10**	**12**	**17**

- Now we complete the process, and the array is sorted.

Furthermore, we can use binary search to find the proper location to insert the selected item at each iteration.

(a) [5 marks] Write a binary search based function to find the position where item should be inserted in a[low..high].

int binarySearch(int a[], int item, int low, int high)

Your function should take in an array of integers **a**, two position indicators **low** and **high**, as well as an integer **item**, which should be inserted in a[low..high].

(b) [9 marks] Make use the binarySearch function to write a function.

void insertionSort(int a[], int n)

that can sort an array of integers based on the idea of insertion sort. Your function should take in an array of integers **a**, as well as an integer **n**, which represents the size of the array.

=== END OF PAPER ===